

MOOJIT NETWORKS

# DataMover Network

---

## User Guide

**DM Development Group, v1.3.1**

**1/1/2002**

DataMover Network user guide instructions.

## Contents

Introduction .....	6
Screen Shots.....	6
Windows Client .....	6
Windows Server .....	7
Linux Server.....	7
Quick Start Guide .....	7
Theory of Operation.....	8
Client and Server Maximum Connection Limits.....	9
I/O Directions .....	9
IN Direction .....	9
OUT Direction.....	10
BOTH Direction .....	10
LOOPBACK Direction .....	10
Client Operation.....	11
Source Interface Selection .....	11
Multiple Node Selection .....	11
Menu Items.....	11
License Import.....	11
License Register .....	11
License UnRegister .....	11
Options Import Data File.....	11
Options Performance Test .....	11
Options Quick Performance Test .....	11
Options Custom Tests Random I/O Test.....	12
Options Patterns Reset .....	12
Options Patterns Random.....	12
Options Preferences .....	12
View Performance History Bytes Out (Red).....	12
View Performance History Bytes In (Yellow) .....	12
View Performance History Bytes Total (Green).....	12
GUI Items .....	12

Address Type.....	12
Parameters and Metrics.....	13
Performance Real-Time Graph .....	14
IOPS Real-Time Graph.....	14
Server Listing.....	14
Stop All .....	16
Stop .....	16
Help .....	16
Error Checking.....	16
Multiple Nodes.....	16
Windows Server Operation.....	16
Menu Items .....	16
License Import.....	16
License Register .....	16
License UnRegister .....	16
Options Preferences .....	16
GUI Items .....	17
Interface.....	17
Active Connections .....	17
Listening On .....	17
Start.....	17
Stop .....	17
Cancel.....	17
Help .....	17
Client Listing.....	17
Linux Server Installation.....	18
Linux Server Operations.....	19
Menu Items.....	19
Exit.....	19
Stop All Threads .....	19
Start All Threads.....	19
Unregister License.....	20

Register License .....	20
Import and Register License .....	20
Change Listening Interface.....	20
Refresh Display.....	20
Command Line Options.....	20
License Operations.....	20
HASP USB Key License Procedure .....	22
Server License Key Procedure .....	22
Host Firewall Considerations .....	23
Windows Workstation Considerations .....	23
Multiple Node Dialog (MND) .....	23
GUI Items .....	24
Source .....	24
Destination.....	24
Nodes .....	25
Source and Destination Listing.....	25
Add .....	25
Remove .....	25
Clear .....	25
OK.....	25
Cancel.....	25
Functional Tests .....	25
General Tests .....	25
Custom Tests.....	26
Random I/O Test .....	26
Performance Tests .....	27
Introduction .....	27
Theory and Operation.....	27
Quick Performance Test.....	28
Quick Performance Test Results .....	29
Long Performance Test .....	30
Long Performance Test Results.....	31

DataMover Preferences .....	33
DataMover Network Client Preferences .....	33
Network Settings.....	33
Advanced I/O Settings.....	34
Reset .....	34
OK.....	34
Cancel.....	34
DataMover Network Server Preferences.....	34
Network Settings.....	35
OK.....	35
Cancel.....	35
Performance Workload Monitor .....	35
Menu Items.....	37
View Performance History Bytes Out (Red).....	37
View Performance History Bytes In (Yellow) .....	37
View Performance History Bytes Total (Green) .....	37
View Graph Type Standard .....	37
View Graph Type per CPU Utilization .....	37
View Graph Type CPU vs. Transfer Size .....	38
GUI Items .....	38
Top Chart.....	38
Bottom Chart .....	38
Refresh Rate.....	38
Workload Completion.....	38
Current Workload .....	38
OK.....	38
UDP Protocol Operation .....	38
Glossary of Terms.....	38

## Introduction

DataMover Network (DMN) is a native graphical user interface (GUI) host based application with no JAVA or .NET components. DMN are two separate host applications – one providing the server functionality and the other the client functionality. On Linux, DMN provides a command line server application that is compatible with the Windows client and provides the same functionality as the Windows server.

DMN's main purpose is to generate IP4 and IP6 traffic over TCP to networking interfaces to validate and characterize the subsystems that provide these resources. DMN is available in 32bit and 64bit versions. DMN will not bypass any native in-box components provided by Microsoft (MS) or a third-party vendor used to enable these resources on the host.

DMN provides functional and performance workload testing. Functional tests are feature rich, with data integrity, bus triggers, and windows event logging. Advanced performance tests characterize twenty different transfer sizes in three basic I/O directions.

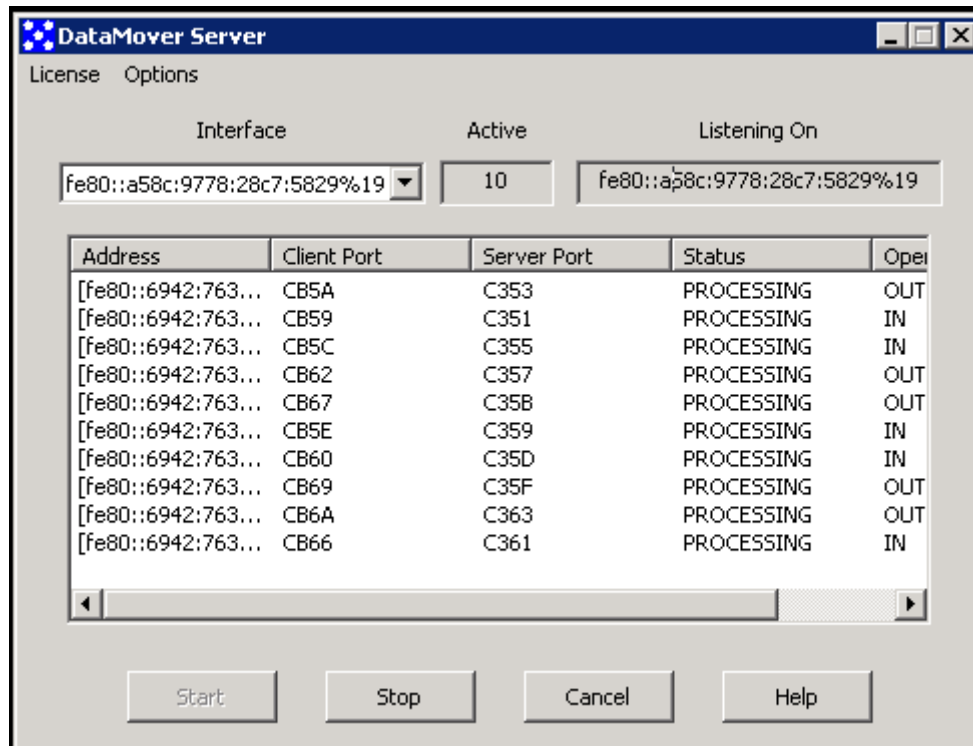
## Screen Shots

### Windows Client

The screenshot displays the DataMover Client application window. The interface includes a menu bar (License, Options, View), a 'Connect' button, and a text field for the address (fe80::a58c:9778:28c7:5829). The 'Address Type' section has radio buttons for IPv4 and IPv6. The 'Parameters and Metrics' section contains several controls: Sessions (5), Transfer Size (1MB), I/O Direction (BOTH), Elapsed Time (0:00:02:39), Status (RUN), and Extended (NONE). Performance metrics are shown as 1196 MB/s and 1055 IO/s. Two performance graphs are visible: 'Performance' with a value of 1334 and 'IOPS' with a value of 2598. A table at the bottom lists active connections with columns for Address, Client Port, Server Port, Status, Operation, MB/s, and IOPS. At the bottom of the window, there are checkboxes for 'Error Checking' (checked) and 'Multiple Nodes' (unchecked), and buttons for 'Stop All', 'Stop', and 'Help'.

Address	Client Port	Server Port	Status	Operation	MB/s	IOPS
[fe80::a58c:9778:28c7:...	CB59	C351	PROCESSING	OUT	114	114
[fe80::a58c:9778:28c7:...	CB5A	C353	PROCESSING	IN	139	97
[fe80::a58c:9778:28c7:...	CB5C	C355	PROCESSING	OUT	118	118
[fe80::a58c:9778:28c7:...	CB5E	C359	PROCESSING	OUT	127	127
[fe80::a58c:9778:28c7:...	CB60	C35D	PROCESSING	OUT	123	123

## Windows Server



## Linux Server

```
[root@license cli]# moojitserver

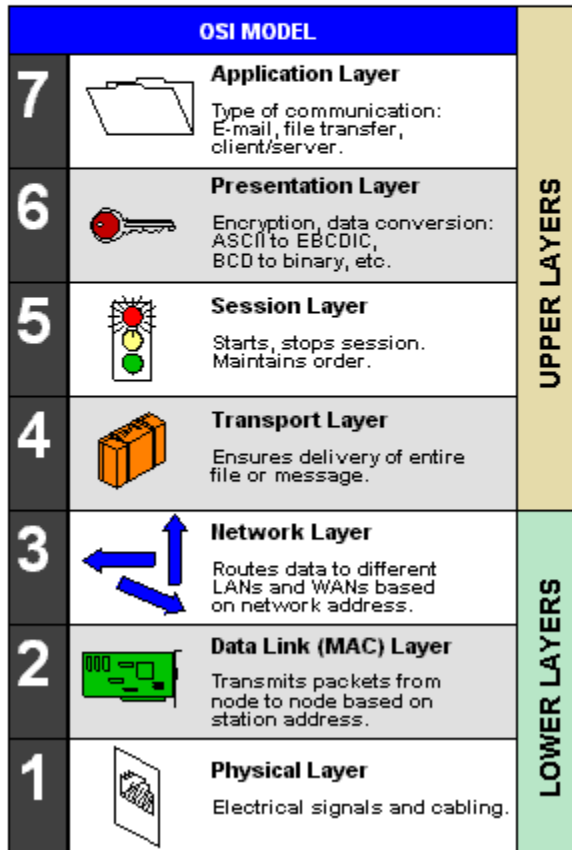
DataMover For Linux Server (32bit) Version 1.0
Registered on license number 264968537
Listening on 193.168.100.164
Active Connections = 0

x - Exit.
s - Stop All Threads.
g - Start All Threads.
u - Unregister License.
r - Register License.
i - Import and Register License.
c - Change Listening Interface.
w - Refresh Display.

>
```

## Quick Start Guide

Download the latest version of DM from [www.moojit.com](http://www.moojit.com) , and install the application.



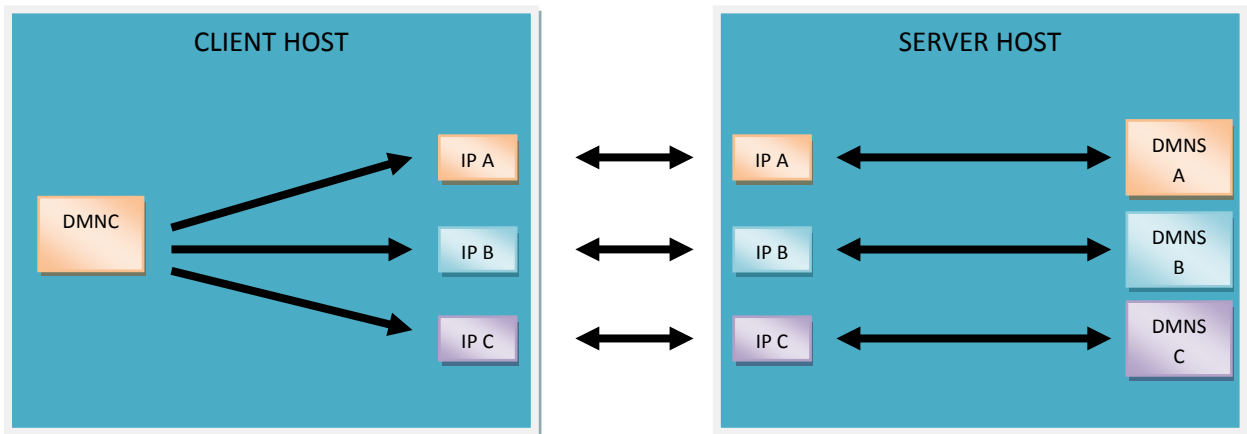
The DMN Server (DMNS) must be up and running first to allow any DMN Client (DMNC) operations to begin. Also, two separate host machines are required – one acting as the DMNS, and the other the DMNC. On the server host, start the DMNS which will automatically bind and start listening on a host network interface. Record the host server IP addresses displayed in the ‘Listening On’ window. This IP will be needed on the host client machine. On the client host, start DMNC and enter the DMNS IP address into ‘Address Type’ section and select ‘Connect.’

By default, the DMNC will begin I/O operations with the DMNS using an automatically selected client host network source interface. The default I/O direction will be ‘IN’ with a transfer size of 1 MiB, data pattern 0xAAh and error checking disabled.

### Theory of Operation

The DMNS and DMNC support IP4 and IP6 addressing and Transport Control Protocol (TCP), User Datagram Protocol (UDP) is not supported. Referencing the Open Systems Interconnection (OSI) Model, DMN exercises all seven layers.

Each connection between the DMNC and DMNS operate on a unique IP & TCP port combination. Each instance of DMNS listens on only one host interface. If the user wants DMNS to listen on multiple host interfaces simultaneously, multiple instances of DMNS must be started on the host. This is not true, however, of DMNC which can connect to multiple DMNS addresses from the same instance, see below.





## Client and Server Maximum Connection Limits

The DMNC and DMNS do not support a limitless number of TCP connections. The DMNS can support 150 TCP connections per direction. For example, the DMNS supports 150 IN and 150 OUT connections simultaneously, providing a total of 300 simultaneous TCP connections. The DMNC supports more TCP connections because it has the ability to talk IP4 and IP6 simultaneously. The DMNC supports 150 TCP connections per direction per IP4 or IP6 address. For example, the DMNC supports 150 IN IP4 and 150 IN IP6 and 150 OUT IP4 and 150 OUT IP6 connections, producing a total of 600 simultaneous TCP connections.

	Client	Server	Direction	Comments
IP4	150	150	IN	
IP4	150	150	OUT	
IP4	300	300	BOTH	
IP4	150	150	LOOPBACK	
IP6	150	150	IN	
IP6	150	150	OUT	
IP6	300	300	BOTH	
IP6	150	150	LOOPBACK	
IP4 & IP6	300	N/A	IN	The client can talk IP4 and IP6 at the same time, while the server cannot. This combination requires two instances of server running.
IP4 & IP6	300	N/A	OUT	The client can talk IP4 and IP6 at the same time, while the server cannot. This combination requires two instances of server running.
IP4 & IP6	600	N/A	BOTH	The client can talk IP4 and IP6 at the same time, while the server cannot. This combination requires two instances of server running.
IP4 & IP6	300	N/A	LOOPBACK	The client can talk IP4 and IP6 at the same time, while the server cannot. This combination requires two instances of server running.

## I/O Directions

Four basic I/O directions are used to define traffic between the DMNC and DMNS – IN, OUT, BOTH and LOOPBACK. Traffic direction is always defined from the perspective of application instance. For example, when DMNC I/O direction is IN, the DMNS I/O direction for the same traffic will be OUT or opposite of the client direction.

### IN Direction

From the DMNC perspective, IN describes traffic that is transmitted by the DMNS to the DMNC. From the DMNS perspective, IN describes traffic that is transmitted by the DMNC to the DMNS.



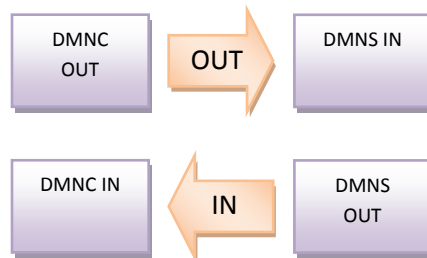
### OUT Direction

From the DMNC perspective, OUT describes traffic that is transmitted by the DMNC to the DMNS. From the DMNS perspective, OUT describes traffic that is transmitted by the DMNS to the DMNC.



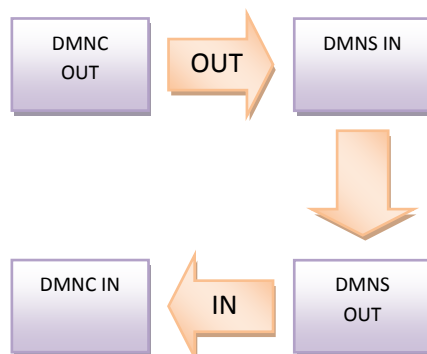
### BOTH Direction

The BOTH direction defines I/O that is occurring in the IN and OUT directions simultaneously (a.k.a full-duplex mode). Traffic that is only occurring IN or OUT is also known as half-duplex mode. When this I/O mode has been selected, both DMNC and DMNS will be transmitting data.



### LOOPBACK Direction

The Moojit would like to think that this I/O direction is unique to DataMover and has not been replicated elsewhere in the dark abyss of open source mediocre I/O generation applications. In LOOPBACK mode, the DMNC sources or transmits the data and the DMNS reflects or re-transmits the data back to the DMNC. The Moojit thinks of this as an external loopback test with a triple shot of



espresso!

## **Client Operation**

The DMNC is a control freak, always dictating the type of I/O that will occur between it and the DMNS. Connection setup and initialization is accomplished by using a management channel between the DMNC and DMNS that allows them to exchange control and status messages.

## **Source Interface Selection**

When the 'Enable Source IP' checkbox is checked under the Advanced I/O Settings in the DataMover Network Preferences Dialog (DMNPD), the user will be prompted to select a DMNC source interface for new connections. By default, this setting is disabled and DMNC allows the operating system to automatically select the DMNC source interface.

## **Multiple Node Selection**

When the 'Multiple Nodes' checkbox is enabled, the Multi-Node Dialog (MND) will be displayed when the user starts a new connection request with the DMNS. By default, this feature is disabled. The MND gives the user the flexibility to create one or more source/destination IP pairings that will be used for this connection request.

## **Menu Items**

### **License Import**

This selection imports the license file. This procedure is not required if using the license USB key and is only required when using the licensing server method.

### **License Register**

This selection registers the application with the licensing server. Use this selection to renew the license lease. This procedure is not required if using the license USB key and is only required when using the license server method.

### **License UnRegister**

This selection removes the registration from the licensing server. Use this selection when preparing to move a license seat to a new machine. This procedure is not required if using the license USB key and is only required when using the licensing server method.

### **Options Import Data File**

This selection imports a custom data pattern.

### **Options Performance Test**

This selection enables the long performance test. Duration: > 1 hr.

### **Options Quick Performance Test**

This selection enables the quick or short performance test. Duration: < 1 hr.

### **Options Custom Tests Random I/O Test**

This selection enables the automated random I/O test. See custom tests section for more information.

### **Options Patterns Reset**

Sets the pattern back to 0xAAh, "AA" will be displayed in "Pattern" window control.

### **Options Patterns Random**

Enables random data patterns, "RND" will be displayed in the "Pattern" window control.

### **Options Preferences**

This will open the DataMover Preferences Dialog (DPD) that allows the user to adjust specific settings related to internet access for registration purposes and advanced features.

### **View Performance History Bytes Out (Red)**

Enable or disable OUT Performance and IOPS graph data.

### **View Performance History Bytes In (Yellow)**

Enable or disable IN Performance and IOPS graph data.

### **View Performance History Bytes Total (Green)**

Enable or disable Total Performance and IOPS graph data. This setting is enabled by default.

## **GUI Items**

### **Address Type**

The address type section allows the user to enter the destination IP4 or IP6 address of the DMNS and start I/O.

### ***Connect***

This button starts the I/O session(s) with the server

### ***IP4 Radio Button***

Radio button control, when selected, enables the IP4 addressing mode

### ***IP4 Entry Window***

The IP4 address of the server, this field will be enabled only when the IP4 radio button is selected

### ***IP6 Radio Button***

Radio button control, when selected, enables the IP6 addressing mode

### ***IP6 Entry Window***

The IP6 address of the server, this field will be enabled only when the IP6 radio button is selected.

## Parameters and Metrics

### *Sessions*

A dropdown control allowing the user to select the number of TCP session's to open with the server for the next connection request

### *Transfer Size*

A dropdown control providing selectable transfer size's, ranging from 8B to 4MiB

### *I/O Direction*

A dropdown control that allows the user to select one of four I/O directions

- IN - Half duplex read operation where server transmits data to the client using the predetermined transfer size
- OUT - Half duplex write operation where the client transmits data to the server using the predetermined transfer size
- BOTH - Full duplex operation where the server transmits data to the client and the client transmits data to the server simultaneously at the predetermined transfer size
- LOOPBACK - Sequential half duplex operation where the client transmits data to the server at the predetermined transfer size and waits for the server to transmit the same data back to the client at the predetermined transfer size. This process repeats indefinitely

### *Elapsed Time*

Elapsed time in Days: Hours: Minutes: Seconds since all sessions were stopped

### *Status*

Displays one of three general status modes relating to all active client sessions

- RUN - Will be displayed if at least one session is active
- STOP - Will be displayed if no sessions are active
- FAIL - Will be displayed if one or more session threads detects an error (see application event log entry for more detail)

### *Extended*

Displays error messages generated by session threads when they occur (see application event log entry for more details)

### *Performance*

Total aggregate throughput of all active sessions in MiB/s (megabytes per second)

### *IOPS*

Total aggregate IO completions of all active sessions per second (IO per second)

### ***Pattern***

Edit window that allows the user to input a byte pattern that will be used in I/O operations (0x00h to 0xFFh). This pattern will be used to initialize the I/O data buffers. This window will also become read only when random data pattern or import data patterns have been selected

### ***Active Connection***

Number of active sessions on the client

### ***Special***

Displays special test mode being used; currently 'QPERFTEST' for Quick Performance Test and 'PERFTEST' for Long Performance Test will be displayed

### **Performance Real-Time Graph**

The real-time graph displays the total throughput across all active sessions for all I/O directions. Double clicking on the plot area will reset values back to zero. Default display setting set for BOTH I/O direction

### **IOPS Real-Time Graph**

The real-time graph displays the total Input/Output/per/sec (IOPS) across all active sessions for all I/O directions. Double clicking on the plot area will reset values back to zero. Default display setting set for BOTH I/O direction

### **Server Listing**

The server listing provides all of the connection information for each specific session between the DMNC and DMNS.

### ***Address***

IP address of connected server

### ***Client Port***

TCP port of client

### ***Server Port***

TCP port of server

### ***Status***

Status of this connection

1. CONNECTED - TCP connection established with server but not processing I/O yet
2. PROCESSING - Processing I/O with server
3. ERROR - Error has occurred (see Windows Application Event log for more details)
4. DISABLED - No longer connected or processing I/O with server

### ***Operation***

Type of I/O being performed with server

1. IN - Client is receiving data transmitted by the server. If error checking has been enabled, client will perform data integrity test on received data
2. OUT - Client is transmitting data to the server. If error checking has been enabled, server will perform data integrity test on received data
3. BOTH - Client is receiving data from the server and transmitting data to the server simultaneously. If error checking has been enabled, client will perform data integrity test on received data and server will perform data integrity on received data
4. LOOPBACK - Client transmits and receives data to and from the server in a sequential fashion. Data is first transmitted by the client; server receives data and transmits back to client. Client receives same data it transmitted and repeats operation. If error checking is enabled, client performs data integrity test on received data

### ***MB/s***

Aggregate throughput of all active sessions in megabytes per second

### ***IOPS***

Aggregate I/O completions per second for all active sessions

### ***MB/s (IN)***

Individual throughput for active session in megabytes per second (receive only)

### ***IOPS (IN)***

Individual I/O per second for active session (receive only)

### ***MB/s (OUT)***

Individual throughput for active session in megabytes per second (transmit only)

### ***IOPS (OUT)***

Individual I/O per second for active session (transmit only)

### ***Chunk***

Transfer size used in all I/O transaction types

### ***Error Checking***

Will display 'Enabled' or 'Disabled' depending on error checking selection for session

### ***Pattern***

Data pattern or data pattern type being used for sessions

### ***Error***

Error code (see Windows Application event log for more details)

### ***Special***

Displays special I/O mode when selected

## **Stop All**

Stops all I/O on DMNC by ending all connections simultaneously

## **Stop**

Allows the user to stop individual DMNC connections by highlighting one or more entries in the 'Server Listing'

## **Help**

Displays a basic help dialog, please go to [www.moojit.com](http://www.moojit.com) for detailed online help

## **Error Checking**

Check box control that enables or disables data integrity for new I/O that is started

## **Multiple Nodes**

Check box control that enables or disables the DMNC multiple node functionality

## **Windows Server Operation**

The DMNS requires minimal configuration input from the user. When it's started, the DMNS will automatically start listening on the first valid IPv4 or IPv6 interface that it finds on the host system but the user may select a different interface if desired. Changing to a new interface is as simple as selecting a new address in the interface dropdown and stopping/starting the server application.

## **Menu Items**

### **License Import**

This selection imports the license file. This procedure is not required if using the license USB key and is only required when using the licensing server method.

### **License Register**

This selection registers the application with the licensing server. Use this selection to renew the license lease. This procedure is not required if using the license USB key and is only required when using the licensing server method.

### **License UnRegister**

This selection removes the registration from the licensing server. Use this selection when preparing to move a license seat to a new machine. This procedure is not required if using the license USB key and is only required when using the licensing server method.

### **Options Preferences**

This will open the DataMover Preferences Dialog (DPD) that allows the user to adjust specific settings related to internet access for registration purposes.



## GUI Items

### Interface

Dropdown that allows the user to select a new listening interface

### Active Connections

Number of active sessions active on the server that are processing I/O

### Listening On

Current interface that the server is listening and servicing requests on

### Start

Starts the server application on the specified interface

### Stop

Disconnects all active connections and stops the server application from listening or servicing any further TCP processing

### Cancel

Disconnects all active connections, stops the server application and terminates the application

### Help

Displays local help window

### Client Listing

Displays a list of all active and previously active connections

### Address

IP address of client connected to server.

### Client Port

TCP port of client

### Server Port

TCP port of server

### Status

Status of current connection

- LISTENING - Server is waiting for a connection.
- CONNECTED - Server is connected but not performing data I/O with the client.
- PROCESSING - Server is connected and performing data I/O with the client.
- DISABLED - Server is disabled on this specific port. This should only occur if the STOP button has been pressed or a serious error has occurred on the server (see application event log for details).

## **Operation**

Type of I/O operation being performed with the client

- IN - Server is currently receiving data from the client (client is performing a write operation). If error checking has been enabled on the client, the server will perform a data integrity check on the received data. If an error is detected, a data miscompare application event log entry will be posted by the server.
- OUT - Server is currently transmitting data to the client (client is performing a read operation).
- LOOPBACK - Server receives and transmits data to the client in a sequential fashion. After the server receives all of the data, it will transmit the data back to the client, mimicking an external loopback operation.

## **Linux Server Installation**

1. Untar the package ('tar -xvf dmnetwork.tar')
2. Run install script ('./dmnetworkinstall.sh')
3. Type 'moojitserver' from any terminal window to start program

## Linux Server Operations

The Linux version of the server side application has the same feature/functionality as its Windows counterpart. It's a command line application that's capable of accepting command line arguments. The server menu appears after entering at the command line 'moojitserver.' The main menu will display version information, the license registration number (if registered), the IP address it's listening on, and the number of active connections.

```
[root@license cli]# moojitserver

DataMover For Linux Server (32bit) Version 1.0
Registered on license number 264968537
Listening on 193.168.100.164
Active Connections = 0

x - Exit.
s - Stop All Threads.
g - Start All Threads.
u - Unregister License.
r - Register License.
i - Import and Register License.
c - Change Listening Interface.
w - Refresh Display.

>
```

In this example, the server has already been registered and it's listening on IP address 193.168.100.164 with zero active clients. Like its Window counterpart, the server supports IP4 and IP6 addressing and is capable of servicing multiple clients simultaneously. Registration and activation requires a licensing server or HASP key installed in a USB port on the host server machine.

### Menu Items

#### Exit

Exits application

#### Stop All Threads

Stops all server listening threads

#### Start All Threads

Starts server listening on a given IP address

## Unregister License

Removes server registration from licensing server and puts server in demonstration mode. All server threads must be stopped before performing this operation.

## Register License

Registers and activates server license. This can be used if the license has been previously unregistered by the user or if using the network licensing server and the temporary lease has expired.

## Import and Register License

This step must be performed first when registering the server license. A valid license file provided by The Moojit is required. Selecting this option will display a submenu that asks the users to enter the path and filename of the license. For example, if the license 12345678.lic is located in the /usr/temp directory, the user should enter '/usr/temp/12345678.lic.' This step will import and register the license in one step.

## Change Listening Interface

Selecting this input, will display a submenu that allows the user to change the IP interface that the server listens on

## Refresh Display

This option will update the main menu page and is helpful when trying to determine the number of active connections

## Command Line Options

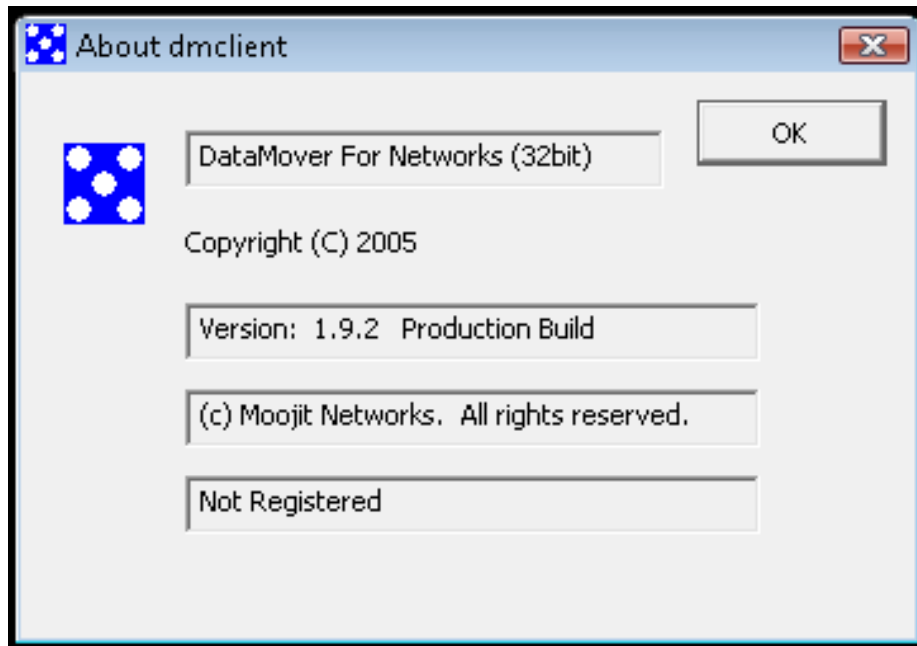
The moojitserver accepts only one command line argument which can be any of the following: IP4 address, IP6 address or interface name. If no command line arguments are provided, the server will be listening on the first valid IP4 or IP6 address that it discovers on the host. If a command line argument is provided and it's a valid IP address or interface name, the server will listen on the interface provided. Below are examples of valid and invalid command line arguments:

- 'moojitserver eth0' - valid
- 'moojitserver xyz' - not valid
- 'moojitserver 192.168.0.1' - valid
- 'moojitserver 2002:1234::1' - valid
- 'moojitserver 192.168.1' - not valid

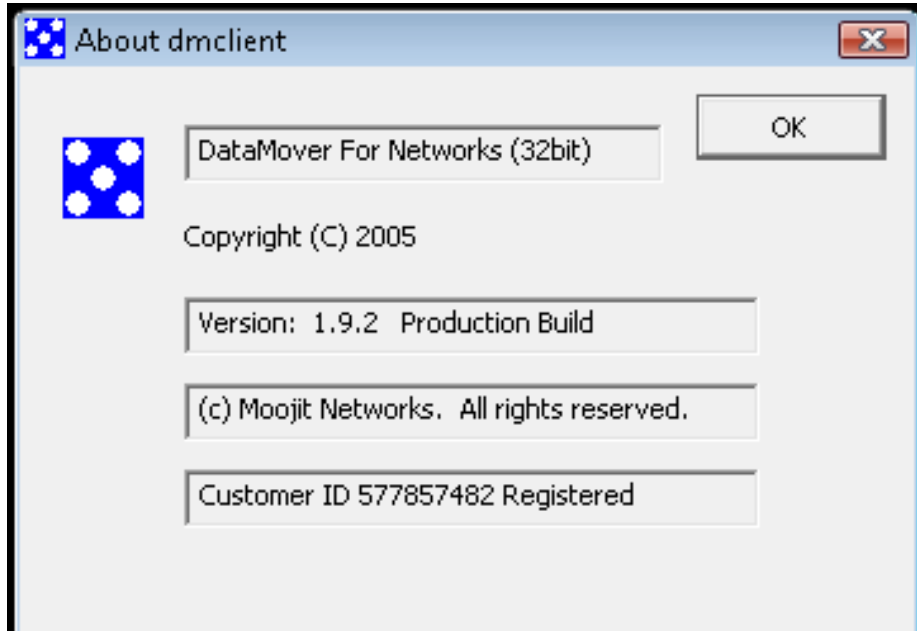
## License Operations

There are two licensing methods DM supports – license server method and HASP USB license key method. If using the licensing server method, a license key must be imported for activation, this step is not required if using the USB license key method. To determine if a license has been activated, open the “About dmclient” or “About dmserver” menu items from the DMNC or DMNS. Both licensing methods require the machine to have at least one network interface card (NIC) enabled.

Screenshot before license activation:



Screenshot after license activation:



Notice how the bottom control change's, displaying the license identification number.

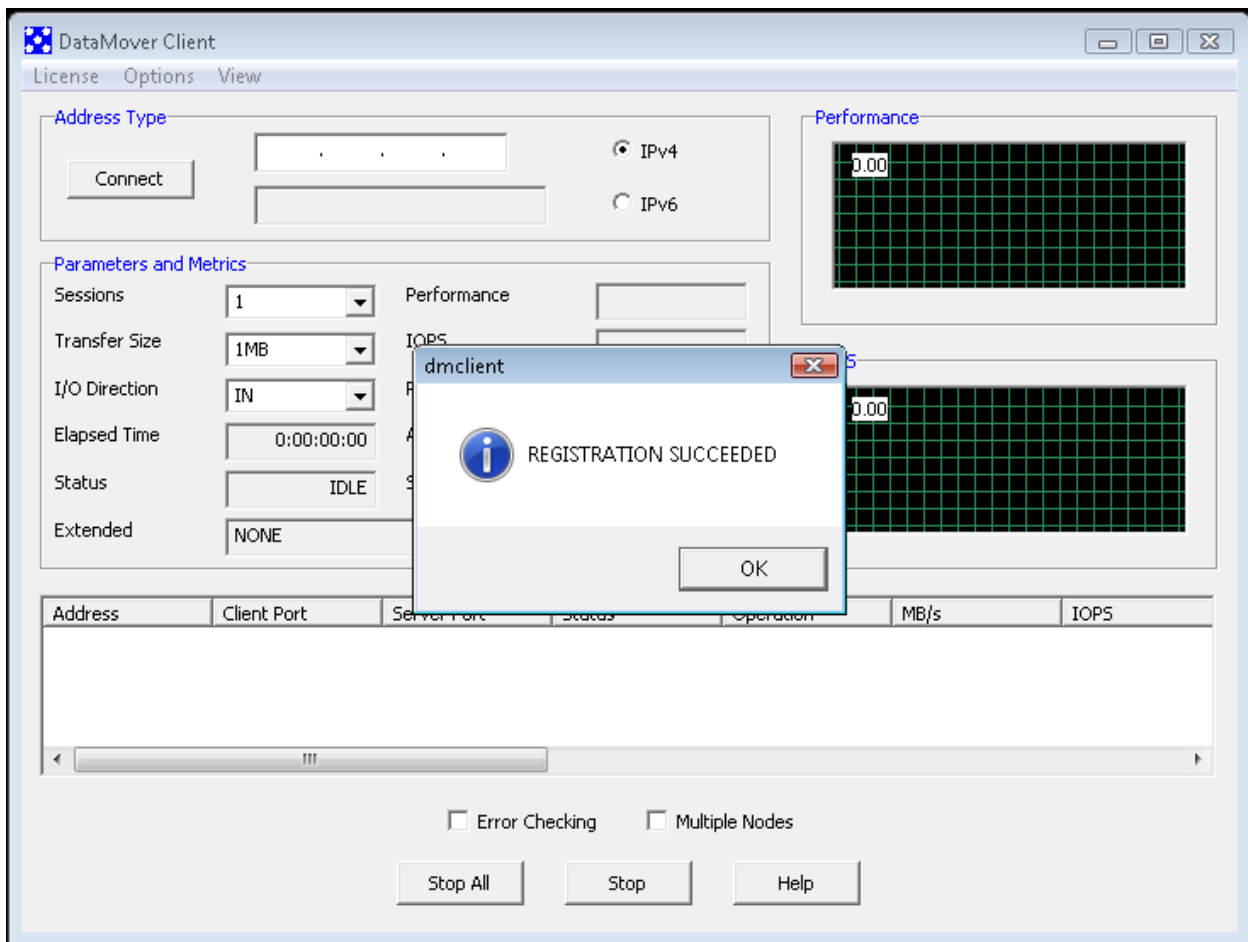
## HASP USB Key License Procedure

The license activation is simple. After installing the application, install the HASP key before starting the application. You're done! The time remaining on the key will be posted in the windows application event log each time the application is opened. The key may be used on multiple machines, move the key to a new machine to activate the DMNC or DMNS.

## Server License Key Procedure

Perform the license import procedure using the "License->Import" menu item. Using the dialog, point to the license file location and select "OK."

Successful license key activation response:



When a license is registered, it will be given a 10 day lease for that particular machine. When the lease expires, the user must perform a license registration by selecting the "License->Register" menu item in the DMNC or DMNS. This will renew the lease for another 10 days on this machine. If the user wishes to move the license from one machine to another, perform the following steps:

1. Unregister the license from the current machine by selecting the “License->UnRegister” menu item in the DMNC or DMNS. This will release the license seat back into the general pool managed by the license server.
2. On the new machine, perform a license import if this machine has never been used before. If DM has been used on this machine before, perform a license registration by selecting the “License->Registration” menu item in the DMNC or DMNS. This will allocate a license seat from the general pool managed by the license server to this new machine.

## **Host Firewall Considerations**

The Windows system that hosts the DMNS application should be prompted by the firewall when DMNS starts asking if it’s okay to unblock the application. If this prompt does not occur, the user will need to manually configure the firewall to allow the DMNS to accept unsolicited inbound TCP connections on the ports it listens on. The Moojit recommends that the user create an exception based on the DMNS binary (dmserver.exe) vs. defining specific ports that need to be opened. The DMNC will not require any firewall exceptions since it initiates only outbound connections from the host.

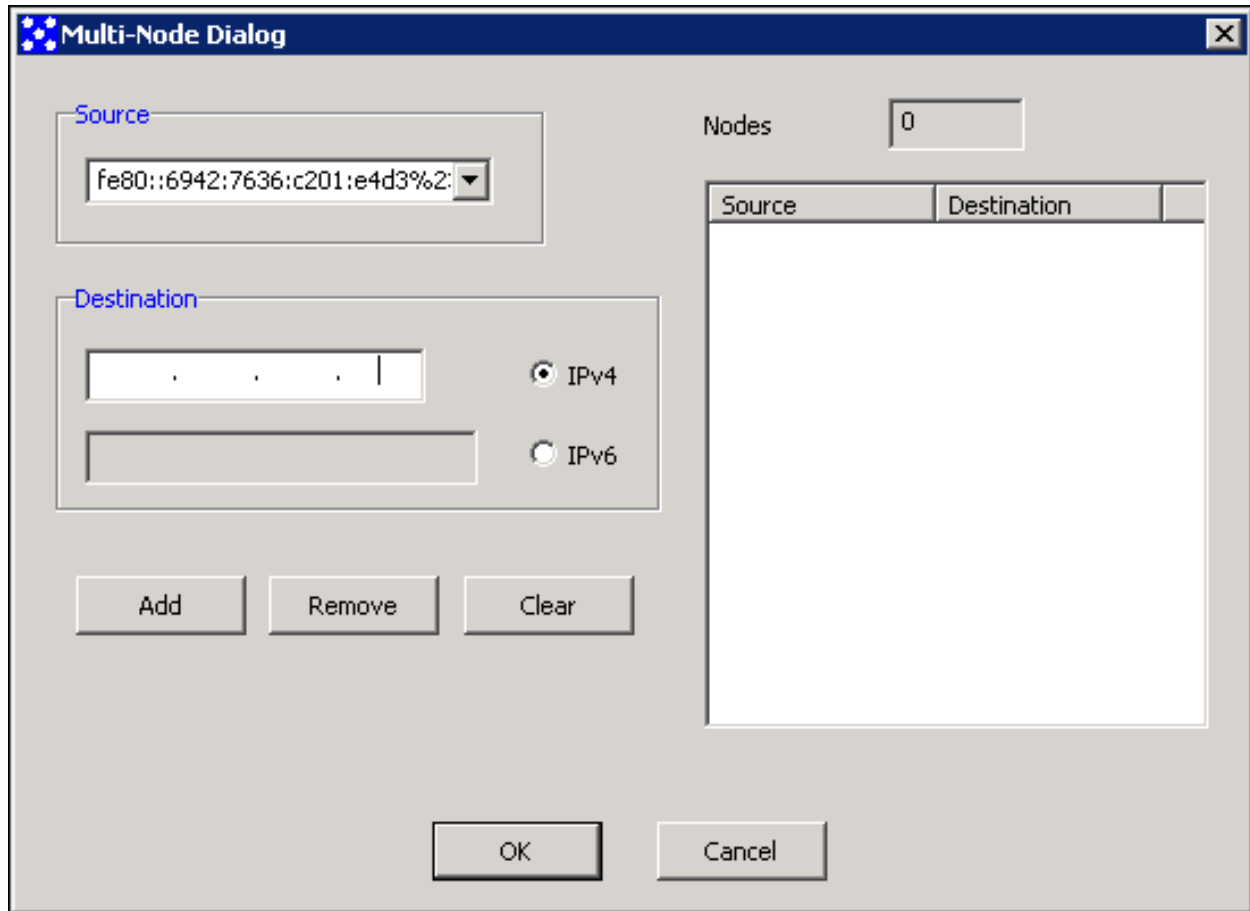
## **Windows Workstation Considerations**

The DMNS should not be run on the workstation or client version of Windows. Microsoft limits the number of inbound connections that can be made. Please refer to Microsoft KB314882 for more information on this topic.

## **Multiple Node Dialog (MND)**

The MND allows the user to create multiple source and destination IP pairings between the DMNC and DMNS.

MND Screenshot:



The pairings will be saved and used each time the user makes a new connection request with the DMNS.

## GUI Items

### Source

A dropdown control that shows all IP4 and IP6 interfaces on the DMNC host.

### Destination

#### *IP4 Radio Button*

Radio button control, when selected, enables the IP4 addressing mode

#### *IP4 Entry Window*

The IP4 address of the server, this field will be enabled only when the IP4 radio button is selected

#### *IP6 Radio Button*

Radio button control, when selected, enables the IP6 addressing mode



### **IP6 Entry Window**

The IP6 address of the server, this field will be enabled only when the IP6 radio button is selected.

### **Nodes**

This control displays the number of entries in the source and destination listing.

### **Source and Destination Listing**

This control displays the source and destination IP combinations that will be used for new connections.

### **Add**

This button adds an entry to the source and destination listing. When selected, the currently selected source interface and active destination interface will become the next entry in the listing.

### **Remove**

This button removes entries from the source and destination listing. Multiple entries may be removed at one time.

### **Clear**

This button removes all entries in the source and destination listing.

### **OK**

This button initiates the connections based on the entries in the source and destination listing and saves the changes.

### **Cancel**

This button cancels the connection request and does not save any changes made to the source and destination listing.

## **Functional Tests**

What attributes defines a functional test and makes them unique? The short answer is error checking.

### **General Tests**

Functional tests can be enabled from the DMNC by checking the “Error Checking” checkbox. When performing functional tests the user must consider the type of test coverage taking place which will be determined by I/O direction (from the DMNC point of reference). The table below explains test coverage based on feature selections.

<b>I/O Direction</b>	<b>Explanation</b>
IN	Each DMNC I/O thread opens a TCP connection with the DMNS and commands the DMNS to begin transmitting data to the DMNC with a pre-defined data pattern and transfer size. Once the transfer size length of data is received by the DMNC, the DMNC performs a data integrity check on the data. This process repeats until the DMNC closes the TCP connection. If data miscompare is detected by the DMNC, the DMNC will log a

	windows application event, display error messages in the DMNC GUI, and close the TCP connection with the DMNS.
OUT	Each DMNC I/O thread opens a TCP connection with the DMNS and commands the DMNS to begin receiving data from the DMNC. The DMNC transmits data to the DMNS using the defined transfer size and data pattern. Once the transfer size length of data is received by the DMNS, the DMNS performs a data integrity check on the data. This process repeats until the DMNC closes the TCP connection. If data miscompare is detected by the DMNS, the DMNS will log a windows application event and close the TCP connection with the DMNC.
BOTH	The DMNC opens two TCP connections with the DMNS for every connection request. One DMNC I/O thread performs I/O in the 'IN' direction; and the other DMNC I/O thread performs I/O in the 'OUT' direction with the DMNS. For the 'IN' direction, the DMNC performs data integrity checks on the received data from the DMNS; for the 'OUT' direction, the DMNS performs data integrity checks on the received data from the DMNC. This process repeats until the DMNC closes the TCP connections.
LOOPBACK	Each DMNC I/O thread opens a TCP connection with the DMNS and commands the DMNS to enter 'LOOPBACK' mode. The DMNC begins transmitting data to the DMNS using at the given transfer size and data pattern. Upon receipt of the data, the DMNS transmits the data back to the DMNC without performing data integrity tests. When the DMNC receives all data from the DMNS, the DMNC performs data integrity checks on the data. This process repeats until the DMNC closes the TCP connection. If data miscompare is detected by the DMNC, the DMNC will log a windows application event, display error messages in the DMNC GUI, and close the TCP connection with the DMNS.

## Custom Tests

### Random I/O Test

This functional test randomly selects workload parameters and periodically updates them on a per connection basis.

I/O Direction	Transfer Sizes	Bus Trigger	Automation	Data Comparison	IP4/IP6	TCP/UDP
Randomly chosen	Randomly chosen	Yes	Full	Always On	Yes, dependent on user connection settings	TCP only

Fixed Settings: Error On, TCP only

Adjustable Settings: Data Pattern, Custom Timer Setting, IP4 only, IP6 only or both, source and destination IP addresses

Purpose: Randomly select I/O parameters with error checking always on.

Automation: The following parameters will be randomly chosen on a per connection basis: I/O direction, number of sessions and transfer size.

Operational Notes: This test will randomly select transfer size and I/O direction, creating unique workload combinations. Adjust the Custom Test Timer in the DMNPD to control the amount of time each iteration runs before a new set of parameters is selected. This test stops only when the 'Stop All' button is clicked. Be sure to check windows application event log for errors.

## Performance Tests

### Introduction

The DMNC supports a quick and long version of the performance test. The quick tests typically last less than 30 minutes, while the long tests may take more than twenty-four hours to complete. The quick tests are not meant to provide the best results, but in some cases they may, their purpose is to provide a short duration test providing a general indication of system performance.

### Theory and Operation

The quick and long performance test characterize network throughput and IOPS for the three basic I/O directions – IN, OUT and BOTH. Twenty different transfer sizes are characterized for each I/O direction ranging from 8B to 4MiB. Unlike DataMover Storage, the DMNC does not vary the workload parameters; instead it uses the number of sessions selected in the 'Sessions' dropdown for all tests. IOPS are not typically associated with network performance characterization but the Moojit believes they are just as applicable. Sometimes the user may notice extraordinarily high IOPS numbers and think there's something wrong with DMNC with respect to how it calculates these values. But this is not the case; the high IOPS are a valid representation of I/O completions from the application's perspective and may also be a valid representation of the hardware's performance.

Measurement results are saved to a comma-delimited file when the test ends. The three basic I/O directions tested are: Half-duplex READ (a.k.a IN), half-duplex WRITE (a.k.a OUT) and full-duplex (a.k.a BOTH). What is meant by half-duplex and full-duplex? Modern I/O technologies support the ability to transmit and receive data at the same time; this is what is meant by full-duplex. Half-duplex implies that only one direction (transmit or receive) is being used at one time. Half-duplex READ means that the DMNC is performing 100% reads from the DMNS on a given TCP connection; half-duplex WRITE means that the DMNC is performing 100% writes to the DMNS on a given TCP connection, and full-duplex means the DMNC is performing reads and writes to the DMNS simultaneously over separate TCP connections. The DMNC provides the following metrics over twenty different transfer sizes ranging from 8B to 4MiB:

- TP (MB/s) - The throughput mean value derived over many samples for given transfer size. For full-duplex, this will be total throughput for both I/O directions.
- TP Standard Deviation (MB/s) – This is the standard deviation of the throughput mean value for a given transfer size. This value will give an indication of variation in data samples.
- IOPS - The IOPS mean value derived over many samples for given transfer size. For full-duplex, this will be total IOPS for both I/O directions.
- IOPS Standard Deviation – This is the standard deviation of the IOPS mean value for a given transfer size. This value will give an indication of variation in data samples.
- CPU (%) - The “CPU” utilization mean value derived over many samples for given transfer size.
- CPU Standard Deviation (%) - The standard deviation of the “CPU” utilization mean value for a given transfer size.

For any given iteration of a performance workload, performance metric measurements will not start until all threads have reached RUN phase. This attribute guarantees that the correct workload will be applied and measured.

### **Quick Performance Test**

The Quick Performance Test (QPT) characterizes two transfer sizes – 8B and 4MiB. The screenshot shows how to enable this test.

The screenshot shows the DataMover Client interface. The 'Options' menu is open, highlighting 'Quick Performance Test'. The 'Parameters and Metrics' section shows 5 sessions, 4MB transfer size, and BOTH I/O direction. Performance metrics include 1294 MB/s and 638 IO/s. A table at the bottom lists connection details for various addresses and ports.

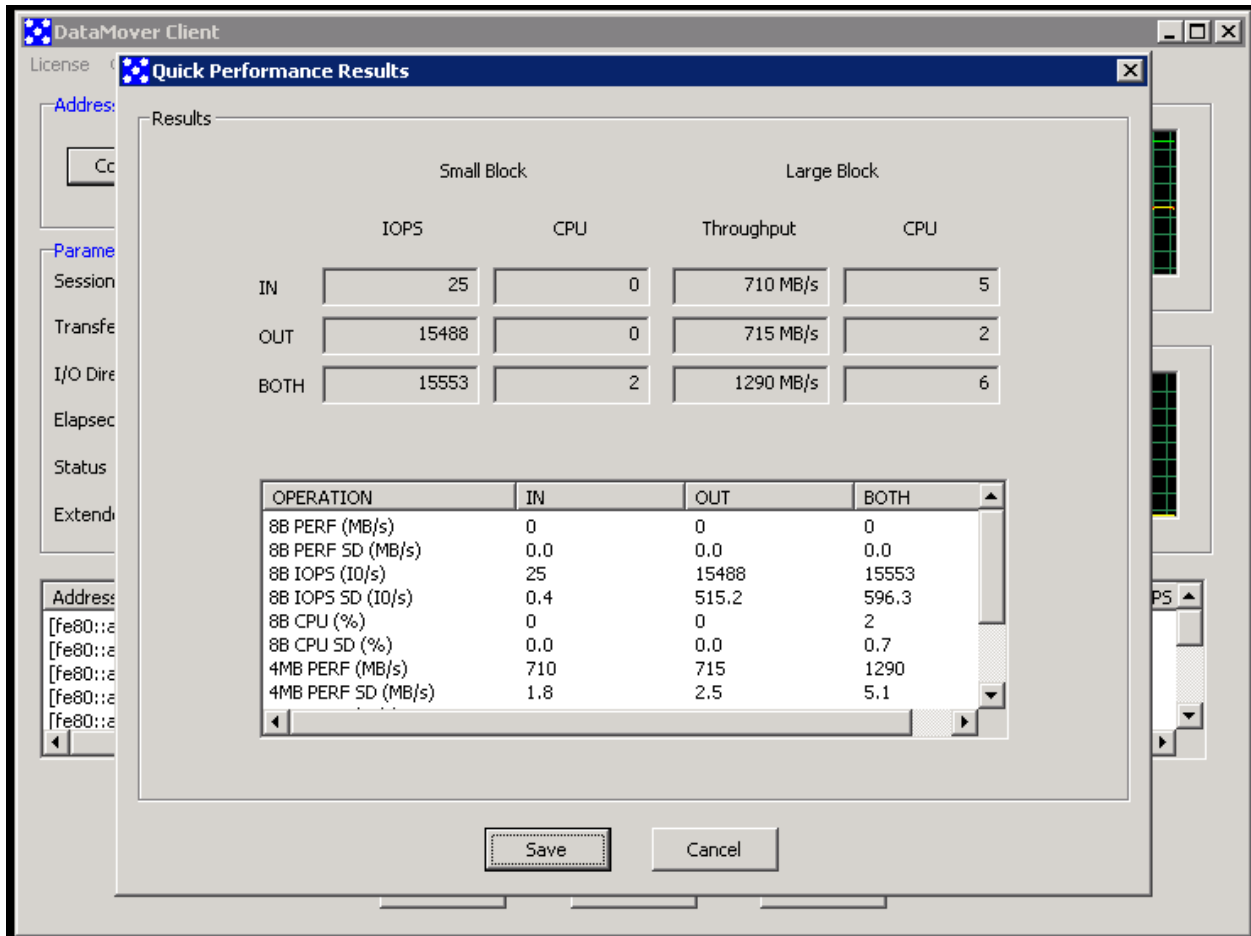
Address	Client Port	Server Port	Status	Operation	MB/s	IOPS
[fe80::a58c:9778:28c7:...] F2FA	F2FA	C35B	DISABLED	IN	127	89
[fe80::a58c:9778:28c7:...] F306	F306	C361	DISABLED	OUT	135	34
[fe80::a58c:9778:28c7:...] F2FC	F2FC	C355	DISABLED	OUT	97	24
[fe80::a58c:9778:28c7:...] F303	F303	C359	DISABLED	OUT	151	38
[fe80::a58c:9778:28c7:...] F2F8	F2F8	C357	DISABLED	IN	131	96

The 'Sessions' parameter will determine how many threads or concurrent TCP connections will be running. For this example, five sessions will be used for the QPT; this produces five threads or TCP connections for the IN I/O direction; five threads for the OUT I/O direction, and ten threads or TCP connections for the BOTH I/O direction.

### Quick Performance Test Results

When the QPT completes, the Quick Performance Test Results Dialog (QPTRD) will automatically display.

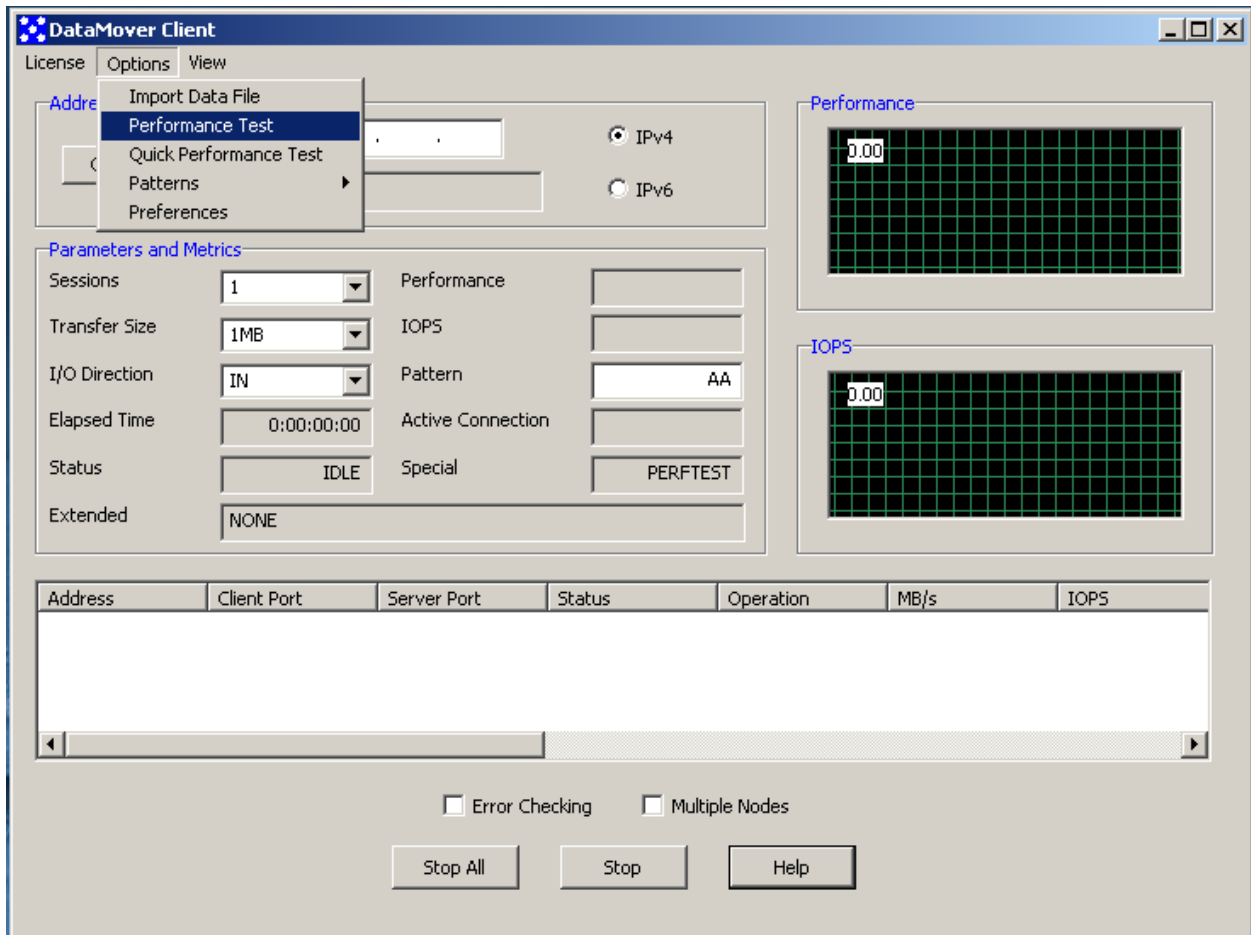
Screenshot:



Selecting "Save," will save the results to a text file not comma-delimited. The results are not intended to be as detailed as the long performance test since this is a quick test. The QPTRD consists of a Small and Large Block section representing data associated with the smallest transfer size of 8B and larger 4MiB. The small block section provides performance data for IOPS and CPU; large block differs slightly by providing throughput instead of IOPS. This distinction is made because users are most likely to be interested in IOPS for the small transfer sizes.

### Long Performance Test

The Long Performance Test (LPT) characterizes twenty different transfer sizes ranging from 8B to 4MiB. This screenshot shows how to enable this test.

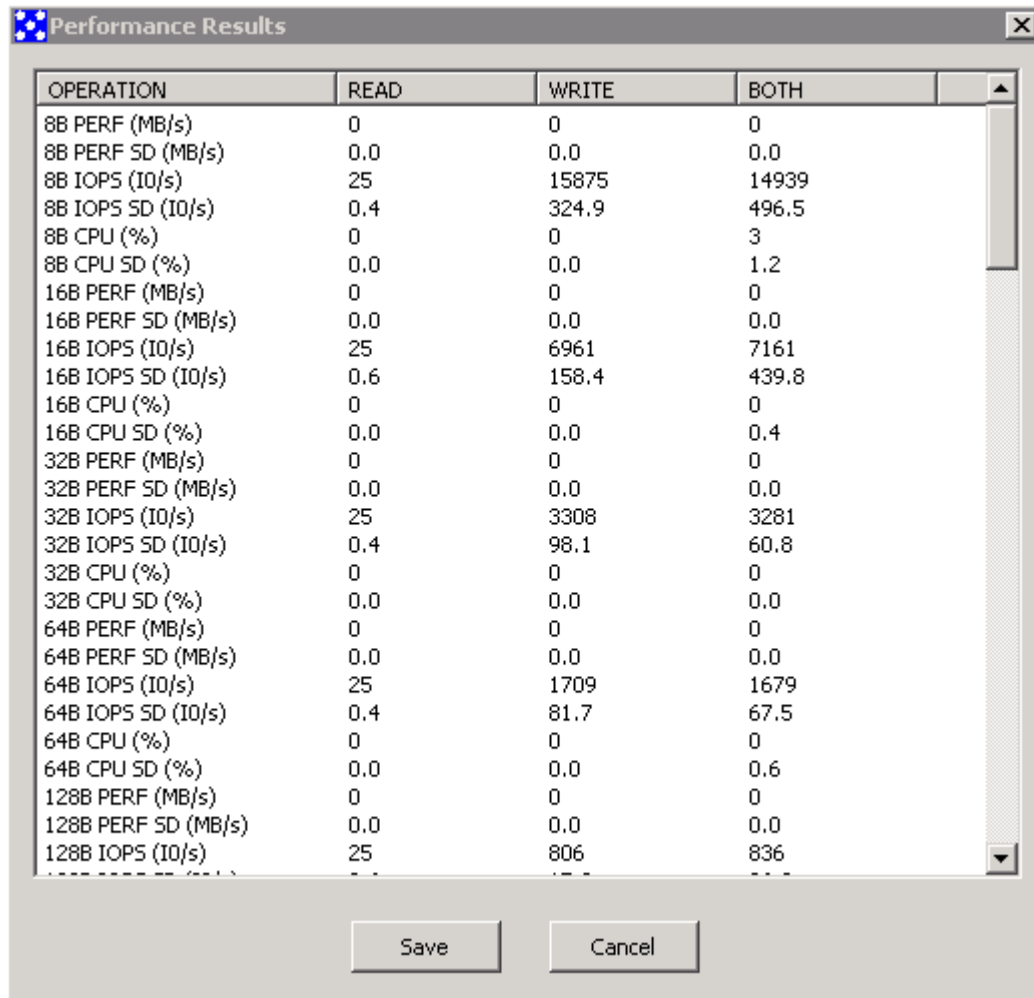


The 'Sessions' parameter will determine how many threads or concurrent TCP connections will be running. For this example, five sessions will be used for the QPT; this produces five threads or TCP connections for the IN I/O direction; five threads for the OUT I/O direction, and ten threads or TCP connections for the BOTH I/O direction.

### Long Performance Test Results

When the LPT completes, the Long Performance Test Results Dialog (LPTRD) will automatically display.

Screenshot:



OPERATION	READ	WRITE	BOTH
8B PERF (MB/s)	0	0	0
8B PERF SD (MB/s)	0.0	0.0	0.0
8B IOPS (I0/s)	25	15875	14939
8B IOPS SD (I0/s)	0.4	324.9	496.5
8B CPU (%)	0	0	3
8B CPU SD (%)	0.0	0.0	1.2
16B PERF (MB/s)	0	0	0
16B PERF SD (MB/s)	0.0	0.0	0.0
16B IOPS (I0/s)	25	6961	7161
16B IOPS SD (I0/s)	0.6	158.4	439.8
16B CPU (%)	0	0	0
16B CPU SD (%)	0.0	0.0	0.4
32B PERF (MB/s)	0	0	0
32B PERF SD (MB/s)	0.0	0.0	0.0
32B IOPS (I0/s)	25	3308	3281
32B IOPS SD (I0/s)	0.4	98.1	60.8
32B CPU (%)	0	0	0
32B CPU SD (%)	0.0	0.0	0.0
64B PERF (MB/s)	0	0	0
64B PERF SD (MB/s)	0.0	0.0	0.0
64B IOPS (I0/s)	25	1709	1679
64B IOPS SD (I0/s)	0.4	81.7	67.5
64B CPU (%)	0	0	0
64B CPU SD (%)	0.0	0.0	0.6
128B PERF (MB/s)	0	0	0
128B PERF SD (MB/s)	0.0	0.0	0.0
128B IOPS (I0/s)	25	806	836

By selecting “Save,” the results may be saved to a comma-delimited file. The results file consists of three sections that report Throughput, IOPS and CPU utilization for each I/O direction.

- Half Duplex Read, Write and Full Duplex TP (MB/s)
- Half Duplex Read, Write and Full Duplex IOPS
- Half Duplex Read, Write and Full Duplex CPU (%)

The Moojit provides a sample excel spreadsheet in the data patterns download on [www.moojit.com](http://www.moojit.com) where the performance results section can be directly copied and pasted in the “data” tab automatically creating performance charts.

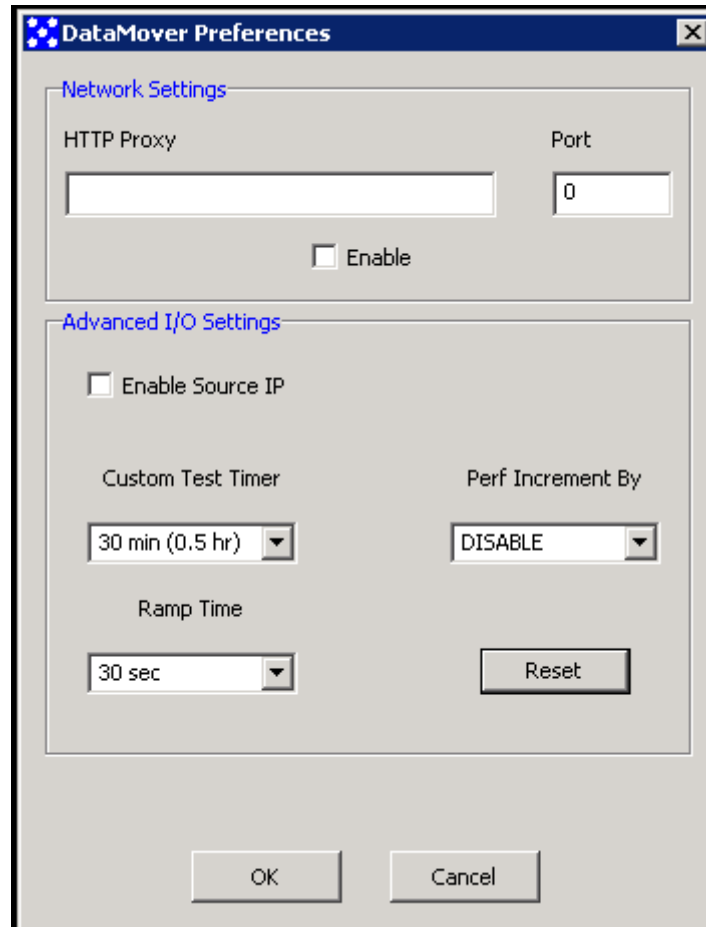


## DataMover Preferences

### DataMover Network Client Preferences

The DataMover Network Client Preferences Dialog (DMNCPD) is displayed from the menu. The DMNCPD allows the user to manipulate Advanced I/O Settings and allows HTTP proxy information to be entered for registration purposes if required.

Screenshot:



### Network Settings

#### *HTTP Proxy*

This is the HTTP proxy URL that must be entered by the user, typically, a DNS name.

#### *Port*

The port in decimal that the HTTP proxy server listens on, typically, port 80.

## **Advanced I/O Settings**

### ***Enable Source IP***

When this checkbox is checked, the user will be prompted to select a source interface for the new connection. When unchecked, the source interface will be selected automatically by the operating system.

### ***Custom Test Timer***

A dropdown control that allows the user to adjust the custom test time ranging from 1 minute to 24 hours in 1 minute increments. The default value is set to 30 minutes.

### ***Perf Increment By***

A dropdown control that allows the user to adjust the performance test session incremental values ranging from 1 to 600. The default value is set to 'DISABLE.' For example, if the number of sessions equals 11 and Perf Increment By equals 2, 6 performance iterations will be performed using a session count of 1 on the first pass, 3 on the second pass, 5, 7, 9 and 11 on the last pass. This setting allows the workload to incrementally increase with each pass. When disabled, only one iteration equal to the number of sessions is used.

### ***Ramp Time***

A dropdown control that allows the user to adjust the performance test iteration initialization time prior to the start of data collection. Values may range from 1 to 300 seconds in one second increments. The default setting is 30 seconds.

### **Reset**

This button control will set all Advanced I/O Settings back to their default values when clicked.

### **OK**

Save the changes made in the DMNCPD and closes the dialog.

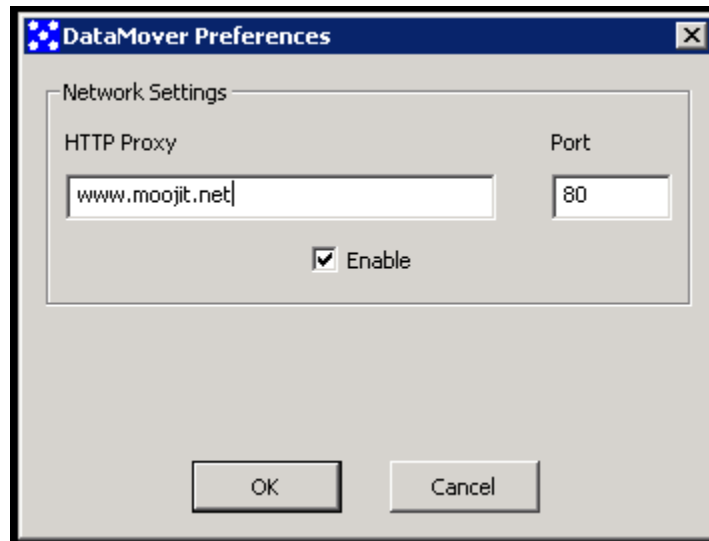
### **Cancel**

Close the DMNCPD without saving the changes.

## **DataMover Network Server Preferences**

The DataMover Network Server Preferences Dialog (DMNSPD) is displayed from the menu. The DMNSPD allows the user to enter HTTP proxy information for registration purposes if required.

Screenshot:



## Network Settings

### *HTTP Proxy*

This is the HTTP proxy URL that must be entered by the user, typically, a DNS name.

### *Port*

The port in decimal that the HTTP proxy server listens on, typically, port 80.

### **OK**

Save the changes made in the DMNCPD and closes the dialog.

### **Cancel**

Close the DMNSPD without saving the changes.

## Performance Workload Monitor

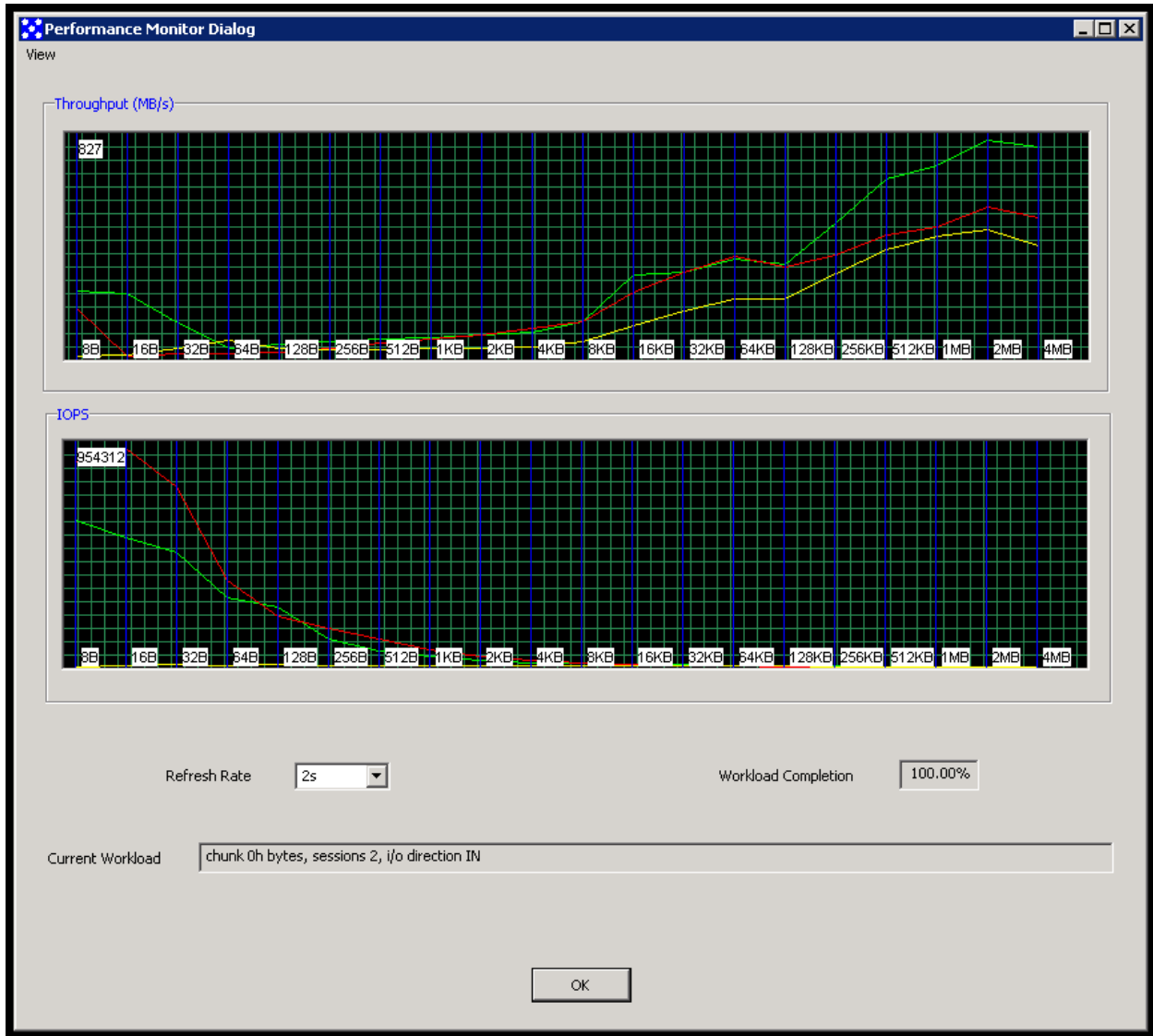
DMNC provides the ability to view the performance metric graphs real-time as the tests are running.

The Performance Monitor Dialog (PMD) allows the user to view any combination of I/O direction and provides the following graph types:

- Standard – Throughput (in MiB/s) and IOPS displayed vs. transfer size for all I/O directions, default I/O direction is READ only.
- Per CPU Utilization – Ratio of throughput and IOPS per CPU utilization displayed vs. transfer size for all I/O directions, default I/O direction is READ only.
- CPU vs. Transfer Size - CPU utilization displayed vs. transfer size for all I/O directions, default I/O direction is READ only.

To access the PMD, select the “Options->Performance Workload Monitor” menu item in the AD. The PMD data will be active only when a performance test is in progress. The PMD will not open if any threads are still in INIT phase.

Default PMD screenshot:



This PMD example shows a test in progress.

## Menu Items

**View Performance History Bytes Out (Red)** - Enable or disable OUT I/O direction graph data.

**View Performance History Bytes In (Yellow)** - Enable or disable IN I/O direction graph data.

**View Performance History Bytes Total (Green)** - Enable or disable Total I/O direction graph data.

**View Graph Type Standard** - Default selection on PMD open. Sets top graph as throughput (in MiB/s) vs. transfer size and bottom graph as IOPS vs. transfer size.

**View Graph Type per CPU Utilization** - Sets top graph as throughput per CPU utilization (in MiB/s/%) vs. transfer size and bottom graph as IOPS per CPU utilization (in IO/s/%) vs. transfer size.

**View Graph Type CPU vs. Transfer Size** - Sets top graph as throughput CPU utilization vs. transfer size and bottom graph as IOPS CPU utilization vs. transfer size. CPU utilization graphs will be the same for throughput and IOPS.

## GUI Items

**Top Chart** – The type of data displayed is determined by graph type selection setting. Default graph type is throughput. Horizontal axis displays transfer sizes ranging from 8B to 4MiB indicated by blue vertical lines. Upper left hand corner displays the maximum value for all data points to provide scale.

**Bottom Chart** - The type of data displayed is determined by graph type selection setting. Default graph type is IOPS. Horizontal axis displays transfer sizes ranging from 8B to 4MiB indicated by blue vertical lines. Upper left hand corner displays the maximum value for all data points to provide scale.

**Refresh Rate** – A dropdown window that allows the user to select the refresh rate of the top and bottom graphs. Possible values are (in seconds) 2s, 5s, 10s, 20s, 30s, 60s, 300s and disable. Default selection is 2s.

**Workload Completion** – Percentage of all workload iterations completed.

**Current Workload** – Displays the current workload parameters being used (transfer size, session count) and I/O direction (IN, OUT or BOTH).

**OK** – Selecting this button closes the PMD.

## UDP Protocol Operation

UDP operation is supported in all modes except for the custom random I/O tests. There is a maximum datagram size that Windows supports. The client and server will negotiate this size during UDP connection initialization using the smallest value reported by either end point. UDP traffic consists of a single packet with a UDP header, if the transfer size is greater than the system's maximum MTU size, subsequent datagram packets will only be enclosed in an L3 header to complete the UDP exchange. If you monitor a DNS server which uses UDP to communicate DNS requests, you will notice that all UDP transfers are less than MTU size and do not require additional L3 UDP packets to complete the transfer. Observations made by The Moojit show that UDP transfers that exceed MTU size will become more unreliable. You may make this same observation by playing around with transfer size while Error Checking is enabled (i.e. data comparison failures will increase with increased transfer size).

## Glossary of Terms

DMNC – DataMover Network Client

DMNS – DataMover Network Server

DMNCPD – DataMover Network Client Preferences Dialog

DMNSPD – DataMover Network Server Preferences Dialog

IOPS – Input/Output Operations per Second

PMD – Performance Monitor Dialog